CHALMERS


# Autonomous Topic-Based Website Categorization

Master's Thesis in Intelligent Systems Design

Golnaz Saberi

Department of Applied Information Technology

CHALMERS UNIVERSITY OF TECHNOLOGY

Technical Report No. 2013:131

ISSN: 1651-4769

Göteborg, Sweden, 2013

Autonomous Topic-Based Website Categorization

GOLNAZ SABERI



Technical Report No. 2013:131

ISSN: 1651-4769

Department of Applied Information Technology

Chalmers University of Technology

SE-412 96 Göteborg

Sweden

Telephone + 46 (0)31-772 1000

# Abstract

Internet has influenced many aspects of our social, economical, educational and professional life. Because of the unique communication means it offers, the internet has grown dramatically since its advent. On the other hand, the ever growing volume of data on the internet has given rise to the demand for establishment of structure on this data. Ranking and indexing web pages by search engines, creation of hierarchical taxonomies of web resources, research on autonomous web page and website classification, are examples of attempts for construction of such structure.

This project includes a study of autonomous website classification. This process has been researched for various purposes and on different levels, especially to improve search engines and directory services. However, the idea of this project comes from a different active area on the internet, i.e. online advertisement. One of the most common sorts of online advertisement are banner ads which are basically published randomly; however, ad servers try to use algorithms to improve the effectiveness of banner ads by publishing them intelligently. One way to do this is to correlate topic of ads and websites they are placed on.

The current project is an attempt towards classification of websites based on their main topic. This work contains a brief study of different web page and website categorization methods conducted to date, as well as implementation of a classification algorithm and analysis of its effectiveness. The implementation consists of creating a graph model of websites and leveraging their link structure for pruning noisy web pages. In addition, a brief description of text classification methods and its relation to the purpose of this project is presented. In this study textual content as well as hyperlink information contained in a website are used to construct a vector space model which is applied for classification by support vector machines (SVM) learning model.

# Acknowledgements

I would like to thank Nils Svangård for giving me the idea of this project and offering me his support. In addition, I want to thank Claes Strannegård, my teacher and examiner, for his understanding and help.

Further, I would like to express my gratitude towards Mr. Mohammad Reza Zeinalizadeh for his invaluable advice which helped me find my way through the implementation of this project.

And last but not least, many thanks to my loving family who are always there for me and bless me with their kindness and support. I also would like to thank my dear friends for being so encouraging and supportive.

# Table of Contents

# List of Tables



# List of Figures

Section One

# Introduction

# 1. Introduction

## 1.1 Motivation

Nowadays, Internet has become the most important source of information for people with different purposes from finding data about any subject to scientific research, to making contact with other people through online communities and social services, to accessing various services, software and online shopping, etc. As the internet expands and the number of users increase the amount of data presented on the web rise up explosively, making the demand for existence of some structure on the web grow dramatically. To establish a structure on the web one way is to create taxonomies of web data which could involve classification of web pages as and websites.

To access the data on the internet, search engines, like 'Google' ("Google, search engine,") and 'Bing' ("Bing, search engine,"), and web directories, such as Yahoo! ("Yahoo!, web directory,") and DMOZ ("DMOZ, open content directory,"), have been designed. Although search engines employ automatic information retrieval methods to index web pages, yet knowledge about functionality of websites could enable search engines to improve the quality of their search results. *Coarse-grained* classification is the area of research that deals with establishing taxonomies of websites with the aim of improving search results' quality. This kind of classification is focused on the *functionality* of websites and creates taxonomies containing very general classes such as 'corporate', 'information', 'nonprofit', etc. Structural properties of a website, like its size, organization and link structure, are the main attributes considered in studies on coarse-grained website classification (Lindemann & Littig, 2007).

Likewise, considering the huge data quantity that is constantly being published on the web, directory services, which are currently developed manually or semi-automatically, would obviously benefit from automation of website categorization. Research focusing on classification of websites to automate the expansion of web directories involves the so-called process of *fine-grained* website classification. Fine-grained website classification is focused on finding the main topic of websites and is more involved with their content (Lindemann & Littig, 2007).

This study contains fine-grained website classification which corresponds to the motivation of the current project, which is improvement of online (or web) advertising. A major part of the Internet, which makes free web services as well as marketing promotion possible, is web advertising. There are different types of advertisements on the web, such as ads by search engines, display ads, mobile ads, classified ads, etc. In

2012 display ads, including banner ads[1], rich media, digital video, etc., constituted 33% of the total advertising on the web, from which banner ads had the greatest share (IAB, 2013).

One of the most important ways that website owners raise money is selling space on their websites for publishing banner ads. Ideally, a banner ad leads the visitor to purchase a product or service, however, given the fact that this is not the case in most of the occasions, banner ads are still highly prevalent because of their success in increasing 'brand awareness' (Dreze & Hussherr, 2003). In any case, the effectiveness of a banner ad can be improved if internet users tend to notice them more often. There are several measures that have been shown to be influential in success of a banner ad, such as the its message and repetition (Dreze & Hussherr, 2003), as well as consistency of the ad with the targeted website's context (Novak & Hoffman, 1997).

Considering the relation between display ads' effectiveness and their placement on different websites, ad servers can profit from knowledge about the purpose of websites in two ways: firstly, if the topic of the website is determined the server can avoid placing ads which are completely unrelated to its context. For example it is much more relevant to show a rubber shoe ad on a sports shopping website than a cooking website. Secondly, users can be clustered based on the websites they are more interested in and the server can utilize these clusters for personalization of ad placement. For example if user 'A' has purchased a product 'p' , then user 'B' who is in the same cluster as 'A', may probably be interested in the same product as well, so the ad for 'p' is a good candidate to be displayed to 'B'. Therefore, ad servers can use this knowledge to enhance display advertising campaigns by intelligently selecting where the ads are to be displayed.

## 1.2 Background

Website categorization has been subject to a number of studies and evolved very much from the time it started. Many of these studies consider a website as one document with its web pages as part of its structure and use classification methods which are based on text classification, and the others regard a website as set of web pages and try to utilize knowledge about taxonomy on web pages in categorizing the website. In the following section a brief discussion is dedicated to previous work on web page classification and the rest consists of examination of related work on website categorization.

[1] Banner ads are one of the primary methods of advertising on the Internet. They are usually clickable small graphics with a very brief advertising message which are displayed on web pages as extra content posted by an external source (Ester, Kriegel, & Schubert, 2002).

### 1.2.1 Web Page Classification

There has been a lot of research on web page classification with different purposes, especially perfecting the search results offered by search engines. Web page classification involves methods for text classification with consideration of the data that can be collected from the structure of web pages, namely html tags, as well as other structural information like the URL, size of the page, links, etc.

Cakrabarti et al. show that categorizing hypertext documents is more difficult than simple text documents, due to high diversity of documents, little consistency in vocabulary, a great portion of documents with very small number of words or those which serve as resource links but at the same time the semi-structured content and existence of connection to other hyper texts can be exploited in categorization of web pages. They improved the simple text classifier by making use of the estimated category of neighboring pages in addition to the page's local text content (Chakrabarti, Dom, & Indyk, 1998). Kwon et al. (Kwon & Lee, 2003) proposed a new weighting scheme for text content of web pages by considering HTML tags and a reformed k-NN classifier for hypertext categorization. In (Sun, Lim, & Ng, 2002) it is shown that SVM classification algorithm is well suited for the problem of web page categorization and using context ( anchor texts plus title tags) data as well as text content dramatically improves the results of classification. In (Yang, Slattery, & Ghani, 2002) they propose that using meta data from the page and its related website dramatically enhances the accuracy of classification. Qi et al. (Qi & Davison, 2006) experimented on this problem with two text classification algorithms and reported SVM to dramatically outperform Naïve Bayes text classifier. However, using their proposed 'neighboring algorithm' with either of the text classifiers has similar results which are better than the results from traditional methods. The neighboring algorithm involves use of information from neighboring pages to find the topic of the target page. Finally, in (Qi & Davison, 2009) emphasizing on the text and labels of co-cited web pages rather than other neighbors and incorporation of anchor texts of parent pages are suggested as important information for classifying a webpage.

### 1.2.2 Website Classification

In this section a number of related studies on website categorization are reviewed.

Ester et al. examined three approaches to categorize websites: 1. Classification of super-pages, 2. Classification of topic vectors, 3. Classification of website trees. The first

method, classifying a website as a super-page, is the simplest one, which is based on extending the procedures which are used for web page classification to categorization of a website. In this way, the whole website is considered as a single feature vector of the frequency of all the words existing in all the pages of the website. This method is simple but offers a very low accuracy. In the next two methods a website is classified based on the topic of web pages constructing it. Firstly, web page topics are determined using naïve Bayes classifier. Secondly, the website is modeled either as a topic frequency vector or a tree of pages with predefined subjects. In case of topic frequency vector representation the classification is performed using C4.5 decision tree classifier. In the second case, Markov chain approach is applied on a 0-Order Markov tree of the website. The website classes considered in the experiments in this study consist of two business classes and the possible page topics are determined manually by studying several websites from each category. This method shows a great improvement compared to the super-page website classification, but the requirement for determination of page topics is both prone to errors and very difficult to perform especially as the number of website classes increase. The best accuracy reached is 87%. (Ester et al., 2002).

Kown et al. continued the same strategy as the above study by adding a new phase to the implementation: web page selection, which is performed by connectivity analysis of web pages of the website. The selected web pages are classified using a probabilistic model which is based on naïve Bayes classifier. They managed to improve performance of classification on a Korean web directory, compared to a previous study which exploits the information gathered from the websites' home page only (Kwon & Lee, 2003).

Kreigel et al. tried to further improve website classification by removing the requirement for labeling web pages to improve the efficiency of training. Here each web page is represented as a feature vector of term frequencies which is used during website classification without being labeled. Websites are presented as sets of feature vectors, and the k-NN classifier is used to predict website classes using SMD (Sum of Minimum Distances) distance measure which is used for calculating distance between set of vectors. Then a centroid set is derived for each website class and a new website is categorized by comparing it to all the centroid sets using half-SMD as distance measure, during an incremental classification approach (Kriegel & Schubert, 2004).

In the next study Tian, et al. extended the tree-based model of website to a page tree of DOM trees. The whole website, the page tree, is represented as an HMT (Hidden Markov Tree) and each page as a DOM tree. In addition, to improve the accuracy and reduce classification overhead a two-phase denoising algorithm as well as entropy based page pruning is implemented. The classification consists of three phases: (1) text-based classification of DOM nodes with k-NN or Naïve Bayes classifier (pre-classification), (2) classification of all pages with HMT-based DOM tree classifier (page classification)

and (3) classification of websites with HMT-based page tree classifier (site classification). They use this procedure to categorize academic websites and report an improvement in accuracy using their proposed method in comparison with the previous models (Tian, Huang, & Gao, 2004).

Dong et al. propose a new approach to include information about the link structure of website not by modeling it as a graph or tree but by incorporating this data, as weighted text features, in the website's feature vector, which in this study is called Hybrid Vector Space Model (HVSM) of website. In this model a website is represented as a topic vector which contains information about link structure (anchor texts) as well as text content of the website. Then these vectors are categorized with a centroid –based classification algorithm. They examine their algorithm for two website categories: Manufacturing and Non-Manufacturing websites and a good performance is reported for the dataset under test (Dong, Qi, & Gu, 2005).

In the next study conducted by Zhang et al. a directed graph is utilized as the base model for a website. They propose a page pruning strategy, in which first the strongly connected component of the graph containing the website's homepage is extracted and then by applying an improved PageRank algorithm on the sub-graph the most topic-related web pages are selected. Lastly, they use the hybrid vector space model to represent the website for classification with support vector machines (SVM). This strategy is used for a 16-class set of website categories and the results are compared to those performed by super-page website classification approach for the same dataset. They report that this results show the proposed algorithm highly enhances the accuracy of website classification (Zhang, Xu, Xiu, & Pan, 2010).

Finally, Lindemann and Littig suggest a new method for coarse-grained website classification which aims at creating taxonomy of all the websites on the internet including a total of 8 classes. They give the general name of super-genre to each of these classes meaning very general categories that are not related to purpose of websites. In this study, many structural characteristics of websites are examined to find a set of them with high information gain. Then this knowledge is leveraged for the first phase of classification of websites into 3 aggregated categories, using naïve Bayes algorithm. To use the textual content of websites, a thesaurus is built for each genre, and once a new website is to be classified, the size of intersection between its word content and every thesaurus is used to measure its similarity to each genre. Lastly, the results of both of the classifications are combined to categorize the website. They report that this algorithm improves the accuracy of coarse website categorization compared to the previous studies (Lindemann & Littig, 2011).

## 1.3 Thesis Outline

The rest of this report consists of three parts which are as follows: section 2, Methodology, section 3, Implementation and section 4, Conclusion. Methodology is composed of four parts including a description of the methods used for modeling websites, web page pruning, feature selection and representation, as well as the classification algorithm utilized in this project. In the next section, Implementation, a detailed record of stages of implementation is presented in two parts, phases and implementation in more detail. The last section, Conclusion, consists of a summary of the project, discussion about the results and finally a brief account about possible future work that can be conducted on this field.

Section Two

# Methodology

# 2. Methodology

This section consists of a description of the website classification algorithm that is used for the purpose of this project.

## 2.1 Website Model

As the summary of the related work shows, website categorization has been studied for very different purposes and datasets of various nature and structure. So it is difficult to decide which method best suits a new purpose and corpus of data. Considering the motivation behind this study I decided to further experiment with the directed graph model of websites, which has shown promising results in a previous study with a similar dataset as the one used in this study (Zhang et al., 2010).

To explain the graph model used in this study we need to define a number of mathematical concepts (Godsil, Royle, & Godsil, 2001):

> Definition 1. A ***directed graph*** $G$ is a set $V$ of vertices and a set $E$ of edges. Each edge is an ordered pair of vertices. Graph $G$ could be defined as a quadruple $G = (V, E, o, t)$ where $o$ and $t$ are mappings $o, t: E \to V$ which relate every edge $e \in E$ to two vertices in $V$. $o(e)$ is called the origin and $t(e)$ the tail of edge $e$.

For a website, each page is a vertex $v \in V$ and a hyperlink between a pair of pages is an edge $e \in E$. A page linking to another one is the origin $o(e)$ of edge $e$ and the tail $t(e)$ would be the page being linked to. Figure 1 shows the graph model of a website.

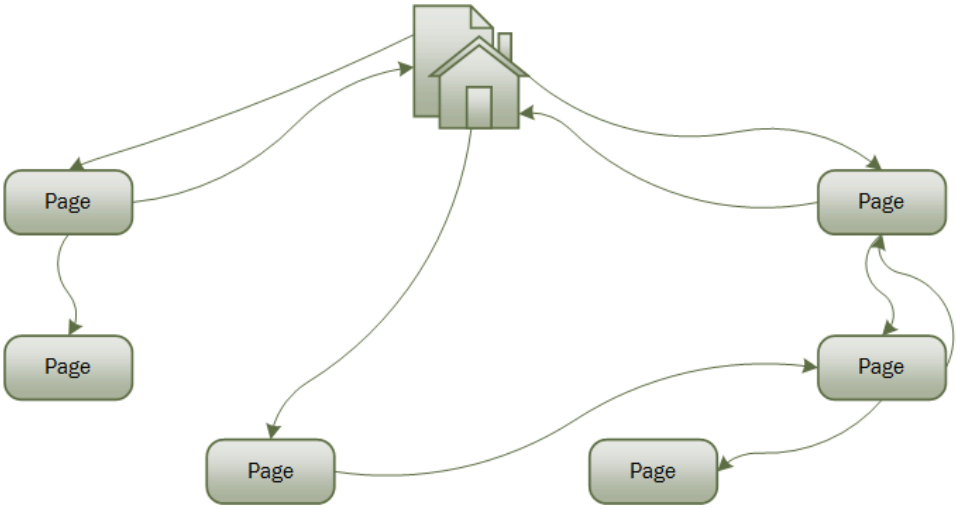


**Figure 1: Website Graph**

## 2.2 Web Page Selection

There are two main reasons that motivate applying an algorithm for page selection. During page selection a number of web pages from a given website are chosen for being downloaded and used for classification. The first motivation for this process is that efficiency of the classification program is dependent on the volume of data being downloaded. The less is the number of downloaded pages from each website, the faster the classification algorithm will be.

On the other hand, as stated in (Ester et al., 2002) the most informative data about the general class of a website resides most probably in pages which are a few links away from the home page. In other words not all the pages of a website contribute to classifying it for many of them act as noise because their content is not consistent with the website's topic. This is the second motivation for page selection.

Page pruning is a subject that has been considered in previous research as well. Ester et al. (Ester et al., 2002) proposed a pruning algorithm based on 0-order Markov tree that can measure the probability of a path being distinctive of a website class. The path includes a number of web pages with known class labels. In (Tian et al., 2004) they use a thesaurus-based text denoising method to prune website's page tree. This method includes determining the coherence of a page topic with the main topic of website by comparing its words to the topic-specific keywords; this method is computationally expensive. Furthermore as the number of topics grows new thesauri need to be built and the efficiency degrades even further.

With reference to previous studies on the structure of websites, there exists a correlation between link structure and content of web pages (Menczer, 2005), and pages that link together are likely to have similar subjects (Chakrabarti, Joshi, Punera, & Pennock, 2002). In another study, Qi et al. (Qi & Davison, 2006) propose a method for web page classification by utilizing information from 'neighboring pages'. Based on the results of their experiments they conclude that knowledge about the neighboring pages greatly improves the classification accuracy. This conclusion also implies that a web page is topically related to the pages in its neighborhood.

Considering the above propositions and study results as well as the fact of inefficiency of content analysis for page pruning, in this study the link structure of a website is used for page selection. As a link between two pages could be a sign that they have related topic, it seems logical to assume that a set of web pages with a compact link structure could represent as the most topic related part of the website graph. To model this set the mathematical concept of *strongly connected component* of the graph is employed (Zhang et al., 2010).

**Definition 2.** A ***path*** $p$ in a graph is a sequence of adjacent edges $p = (e_1, e_2, \dots, e_n)$ where $t(e_{i-1}) = o(e_i)$.

**Definition 3.** A ***strongly connected graph*** is a directed graph in which for any two vertices $v_1, v_2$ there exists a path from $v_1$ to $v_2$ and a path from $v_2$ to $v_1$.

**Definition 4.** A ***component*** of graph $G$ is a connected subgraph $H$ which is not contained in any connected subgraph of $G$ with bigger number of vertices or edges.

In a website, a path from page $x$ to page $y$ consists of a set of hyperlinks which lead from page $x$ to page $y$, through zero or more pages. Further, a strongly connected component represents a set of web pages in the website graph so that there is at least one path from each given page to every other page in the component. With respect to this definition, the strongly connected component of the directed graph in figure 1 is as shown as in figure 2.

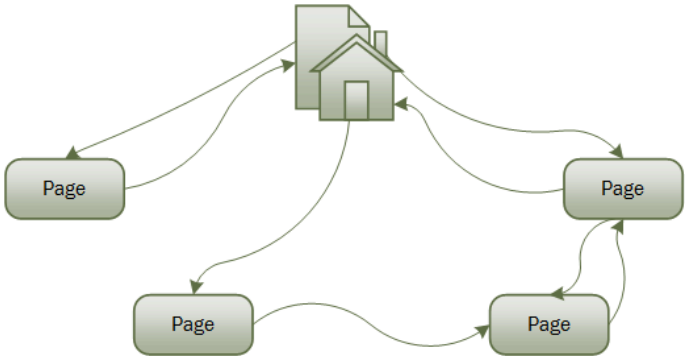


**Figure 2: Strongly Connected Component of Website Graph**

### 2.2.1 Page Ranking

To further refine the set of pages that will be used for classification the PageRank algorithm is employed. PageRank is the hyperlink analysis algorithm used by Google to assign relative weights to all the web pages on the World Wide Web. This weight conveys the relative importance of each web page. This algorithm is based on two assumptions about the importance of a web page (Page, Brin, Motwani, & Winograd, 1999).

1. A link from page A to page B is considered as a vote that contributes to the weight of page B.
2. The more *important* page A is the greater its contribution is to weight of page B which receives a link from A. An important page has generally a large number of incoming links (Page et al., 1999).

Therefore the rank of a web page is influenced by the ranks of all the pages linking to it. Besides, assuming that page $u$ gets a link from page $v$ with rank $R$, if this link is the only link on page $v$, it is intuitively considered more important than if it is one of 100 links on this page. Taking these considerations into account, The rank of a page $u$, $R(u)$ with $B_u$ backlinks (incoming links) is calculated as follows (Brin & Page, 1998):

$$R\ (u) = (1 - d) + d \times \sum_{v \in B_u} \frac{R(v)}{F_v} \qquad \textbf{Equation 1}$$

$R(u)$ or rank of page $u$ represents the probability that a random visitor will visit this page. In the above formula $v$ refers to each website that has a link to page $u$ and $F_v$ as its number of forward links. $d$ is called the damping factor and is usually set to $0.85$ which is the probability that the visitor quits and requests a new page. Page ranks are calculated recursively and will converge after a few iterations. The number of iterations grows with $\log(n)$ where $n$ is the number of pages being ranked (Brin & Page, 1998).

The purpose of page selection in this project is to find the most topic related or similar pages. Zhang et al. (Zhang et al., 2010) propose that a link from a 'similar' page must be more influential in rank of the page receiving the link. They define a simple and intuitive measure for similarity of pages based on their common inner and outer links. However, considering only the first step links, four kinds of link-based relationships between two nodes in a directed graph can be defined:

- There is a link from node $u$ to node $v$; in this case $u$ is called a 'parent' of $v$ and $v$ a 'child' of $u$.

- Nodes $u$ and $v$ link to one or more common node(s); in other words $u$ and $v$ have one or more common child(ren).
- Nodes $u$ and $v$ receive link(s) from one or more common node(s), other than $u$ or $v$, namely, they have one or more common parent(s).

In this study the similarity between two nodes is computed by summing up the total number of common parents and children plus one point for being the other node's child and one point for being the other node's parent. To put the similarity score in effect in rank calculation, this score is multiplied by the rank score in the above formula. In addition, it should be noticed that if a page has links to and from *all* other pages, it will have a high similarity score with all the pages in the graph. On the other hand, if a page has many common links with a few other pages its similarity score to those pages is less than this score for the page in the previous example. To solve this problem, the relative similarity is considered instead of the absolute value. Consequently, $F_v$ , the total number of children of page $v$, in the above formula is replaced by the total value of similarity scores of page $v$ with its children:

$$R\ (u) = (1 - d) + d \times \sum_{v \in B_u} \frac{R(v).\, sim(u, v)}{\sum_{w \in B_v} sim(v, w)} \qquad \textbf{Equation 2}$$

Here $sim(u, v)$is the score given to similarity between page $u$ and page $v$.

Altogether, to select the most topic-related pages, first the strongly connected component of the website graph, which includes the home page, is extracted; secondly, pages in this component are ranked and sorted accordingly. Lastly, a pre-defined number of these pages with the highest ranks are chosen for being used in classification.

## 2.3 Feature Selection and Representation

The next stage, after page selection, is deciding about features that are going to be utilized for classification. Previous studies show that both text and structural features of web pages are proved to be useful for decision making about topic of a website. Various methods have been employed to utilize these features, from two phase classification which separates the process to two stages each including only structural or textual features, to those which incorporate structure in model of the website. In the current study the page selection is based on link structure of the website and the rest of the process is done using a *hybrid vector space model* which incorporates textual content of web pages and a selection of structural features in one topic vector to be leveraged by the same algorithms for text classification (Dong et al., 2005).

### 2.3.1 Text Classification

Due to the increasing volume of digital data that is being available, text classification (TC) has been actively studied in the last 10 years. TC is both used as the process of automatically assigning a label, from a set of predefined set of topics, to a text document (*supervised* approach) or the automatic identification of such a set of topics and grouping documents under them (*unsupervised* approach). TC has been used for different purposes, among which is document indexing for search engines and expanding the web resource hierarchies.

Machine Learning (ML) is the latest and most applied supervised approach in TC. It is based on inductive learning from a set of manually labeled documents. The inductive process, the learner, draws a set of characteristics from the labeled documents, called the training set, to utilize them in classification of an unseen document. In addition, there are two methods for verification of results of classification, (1) *train-and-test*: dividing the corpus of labeled documents to two sets, one for training, called training set, and one for testing, test set; (2) *cross validation*: partitioning the corpus to $k$ sets and iteratively applying train-and-test on the pair $(C - S_i,\ S_i)$ where $C$ is the whole corpus and $S_i$ is the $i^{th}$ set, $1 \leq i \leq k$. In both cases, in every train-and-test process the effectiveness of classification is measured by calculating the number of correctly classified documents in the test set (Sebastiani, 2002).

In general a TC task is composed of several stages:

- *Data preprocessing*. This stage consists of (1) removing all the non-textual elements, like HTML tags, (2) eliminating stop words, words that are common to all kinds of documents, such as ‘is’, ‘have’, etc. and (3) stemming, replacing inflected words with their stems, for instance, ‘libraries’ with ‘library’, ‘talked’ with ‘talk’, etc.(4) removing rare words, word which occur less than a number of times, in this study 3, in the training set.
- *Data representation*: mapping a text into a compact representation that can be interpreted by a classifier. The most common method is using vector-space model that is to present a text as a vector of ‘term weights’.
- *Classification*: the most successful methods for text classification are those which can perform well regarding high dimensionality of vector-space. Naïve Bayes and Support Vector Machines (SVM) are among the most successful approaches in machine learning that are proven to fit TC problems.

### 2.3.2 Vector-Space

In vector-space model, every document is represented by a vector of weights. The elements of this vector, in simple text classification, are most typically words existing in the corpus of training documents. In this study, the structural characteristics that might be drawn from the information contained in HTML documents are also part of the vector-space. These structural data consist of layout information in HTML tags and metadata of website or webpage as well as anchor texts of hyperlinks. Words in the text content of selected web pages are weighted according to the layout or metadata tags they are contained in. On the other hand, anchor texts are, considered as structure features which are compared against similar set of features in other websites. Consequently, in this study the term vector of a website is composed of two groups of features, structure and content features. These two groups are joined together to construct the vector-space of the website.

### 2.3.3 Dimensionality Reduction

The dimension of vector-space is defined by the number of distinct elements contained in structure and content feature sets of all the websites in the training dataset. Usually this number is in the order of tens of thousands of words. But many of these words have very low frequency and are not useful in classification. In addition the efficiency of classification is influenced by dimensionality of vector space. Therefore, often a dimensionality reduction algorithm is employed to reduce the size of feature space.

Reducing the number of features can also help solve the problem of overfitting which happens when the number of features is relatively much bigger than the cardinality of training set. When overfitting happens, the classifier performs very well in reclassifying the training data but badly in classifying unseen instances. This happens because random errors and noisy data are also considered as part of characteristics of the underlying categories.

There are many different methods for dimensionality reduction which can be separated in two groups, term selection and term extraction. For $T$ as the original term set, the result of term selection would be $T'$ which is a subset of $T$; and the effect of term extraction on $T$ would be a set $T''$ in which the members are created by combination or transformation of terms in $T$, which means they are not the same terms any more.

In this study the term selection approach is chosen for dimensionality reduction. The aim of a term selection process is to select a subset $T'$ from $T$ where $|T'| \ll |T|$ where $|T'|$, $|T|$ represent the number of members in $T'$ and $T$ respectively. There are two main

approaches in term selection, wrapper and filtering methods. In the wrapper method a primary term set is chosen randomly and used by the classifier and then the results of applying this set is measured by comparing them to a validation set. This process repeats iteratively by adding or removing words to/from the primary set and in the end the set with best results is chosen as the final term set. Although the selected terms by this method are optimal but it is computationally expensive, therefore not fit for text categorization which usually has very large datasets.

The more efficient alternative for wrapper term selection is filtering method. In this method the importance of words in $T$ are measured based on a predefined criterion and the $n = \frac{|T|'}{|T|}$ words with the highest scores are chosen as the final set. Frequency is the basis of all methods of term selection. The simplest method is choosing words with highest *document frequency (DF)*, the number of documents a term exists in. Term filtering is always performed after removing stop words which have a very high document frequency. So with this method, apart from the stop words, only terms with very low DF are eliminated. On the other hand, based on a well-known law in information retrieval (IR), the words with low to medium document frequency are the most informative ones (Salton & Buckley, 1988). Therefore, DF is a proper choice for term selection because it keeps the most informative terms, and it can reduce the size of term set by a factor of 10 with no loss in effectiveness.

Apart from the simple document frequency method for term selection, other more sophisticated methods have been used for term filtering which can reduce the size of the term set greater with no loss in efficiency. All these functions are created based on the intuitive idea that the best terms are those which can best discriminative of different categories. In other words, the most informative words in category $c_i$ are those which have a high DF in $c_i$ but relatively low DF in other categories. Several functions have been proposed for term filtering and different studies have compared their efficiency on different datasets with various classifiers. One of the functions that can be generalized to multi-class problems and consistently show high efficiency on different datasets is Information Gain (IG) so that it has become a basis for measurement of efficiency of newly proposed filtering algorithms. Particularly, in (Riboni, 2002) which compares the effectiveness of IG and DF for web page classification, IG shows much better performance as the relative size of subset $T'$ reduces. Based on the results of this study performance of IG increases as the size of $T'$ decrease down to $\frac{|T|}{100}$. This algorithm measures the decrease in entropy if a given word is kept in the dataset versus if it is eliminated. For a set of categories $\{c_i\}_{i=1}^{m}$ the following formula shows information gain of term $t$ (Forman, 2003).

$$IG(t) = -\sum_{i=1}^{m} P(c_i)\log P(c_i) + P(t)\sum_{i=1}^{m} P(c_i|t)\log P(c_i|t) + P(\bar{t})\sum_{i=1}^{m} P(c_i|\bar{t})\log P(c_i|\bar{t})$$

**Equation 3**

Here $P(x)$denotes the probability of having $x$ and $P(x|y)$ the conditional probability of $x$ given $y$ and $P(\bar{x})$ is the probability of not having $x$; in the formula of IG the first term refers to the total entropy in distribution of categories, the second term is the entropy in the distribution keeping $t$ and the third one is this entropy when term $t$ is eliminated from the term set.

In addition to IG a newly proposed filtering algorithm called *Distinguishing Feature Selector* (DFS), which is reported to be more effective than IG, is also experimented for term filtering in this study. DFS assigns high score to the terms which are distinctive and low score to irrelevant ones (Uysal & Gunal, 2012). Their proposed algorithm is based on the following requirements:

- An absolutely distinctive word occurs frequently in a single class and not the others.
- A relatively distinctive word occurs in some classes and does not occur in other classes.
- An irrelevant term can be either a term frequently occurring in all classes or one which rarely occurs in one class and does not occur in others.

The following formula represents DFS for term $t$ which embodies the above requirements:

$$DFS(t) = \sum_{i=1}^{m} \frac{P(c_i|t)}{P(\bar{t}|c_i) + P(t|\bar{c_i}) + 1}$$

**Equation 4**

According to the above formula, for a term $t$ that is present in all the members of class $c_i$ and absent in all the other classes, DFS is 1.0 which is the top score. As the occurrence of term $t$ in class $c_i$ reduces the value of DFS also drops. On the other hand, DFS of a word occurring frequently in all of the classes is reduced by including the probability $P(t|\bar{c_i})$ in the denominator of equation 4.

### 2.3.4 Feature weighting

The vector space of a text is composed of a set of term weights representing the *importance* of each term in that specific text. Assignment of proper values to weight of terms in the document's vector plays a vital role in the result of text classification. There are two general methods of weighting: *supervised*, in which the previous knowledge about class of documents is used, and *unsupervised* method which is solely based on the statistics of term in the whole corpus of data. There are various methods for assigning weight to terms, such as the very well-known $tf.idf$ (Jones, 1972) which is the most commonly used method in Information Retrieval.

Term frequency-inverse document frequency, $tf.idf$ is a measure of how important a word is to a document in the dataset. This metric is based on two suppositions; the frequency of a term in a document contributes to its weight, and, the words that are frequent in all the documents of the training set do not discriminate between documents belonging to different classes. Therefore the value of term frequency is multiplied by inverse document frequency ($idf$) to reduce the effect of words that are common in the whole corpus of data. $idf$ for term $t$ in training set $D$ is defined as follows:

$$idf(t, D) = \log \frac{|D|}{|\{d \in D : t \in d\}|} \qquad \textbf{Equation 5}$$

Where $|D|$ is the total number of documents in the training set and the denominator denotes the number of documents $d$ in $D$ which contain word $t$. The more frequent a word is among different documents, the less its value of $idf$ would be. So if $tf$ is equal for two words $t1$ and $t2$ in document $d$, $t1$ which is more contained by other documents in the dataset will have a less weight value compared to $t2$. Of course, if a term is frequent in all the documents of a specific class but not in documents belonging to other classes, it has a high discriminating power. But, as the goal is to classify a new instance, for which the class label is unknown, this knowledge cannot be used to weight the terms.

Further, to consider the layout of Html source of web pages, a weight coefficient is multiplied by tf-idf for every term. The value of this coefficient depends on the tag(s) embodying the term. Weight coefficients are sorted as follows:

$$w_{title} > w_{keyword} > w_{description} > w_{H1} > w_{H2} > w_{H3} > w_{bold} > w_{underlined} > w_{italic}$$

Lastly, as web pages have very different lengths, with a weighting scheme which is solely based on word frequency, those with more words might dominate the knowledge

base. Therefore, the weighting function must remove the effect of length of text, by normalization of topic vector length. In this study, cosine normalization is used to equalize length of documents. A term weight $w_{i,j}$ for term $i$ in document $j$ is assigned as:

$$w_{i,j} = \frac{w_{i,j}}{\sqrt{\sum w_{i,j}^2}} \qquad \textbf{Equation 6}$$

The new weight values are representative of the relative importance of each term in every website.

## 2.4 Classification

Several classification algorithms are known to suit the problem of text classification. Among them the most appreciated are Naïve Bayes, K-Nearest-Neighbor (KNN) algorithm, decision tree classifier and Support Vector Machines (SVM). Different studies have shown that SVM yields promising results in TC tasks; specifically Joachims, et al. (Joachims, 1998) experimentally show this algorithm outperforms the previously mentioned algorithms.

A TC process involves feature sets of very high dimensionality while a large number of the features are relevant. In addition document vectors are sparse, meaning that they have few non-zero entries. SVM is suited for text classification, because it can handle these problems that a TC task involves.

SVM is designed according to the *Structural Risk Minimization* from computational learning theory. The given hypothesis by a learner, which is based on structural risk minimization, must guarantee the lowest true error. True error is the probability that an unseen example is wrongly classified. Support vector machines find hyperplanes that separate training examples with maximum margin, where margin is the distance between the separating hyperplane and the closest training example to it. Figure 3 shows a linear hyperplane in two-dimensional space separating positive and negative examples with maximum margin.

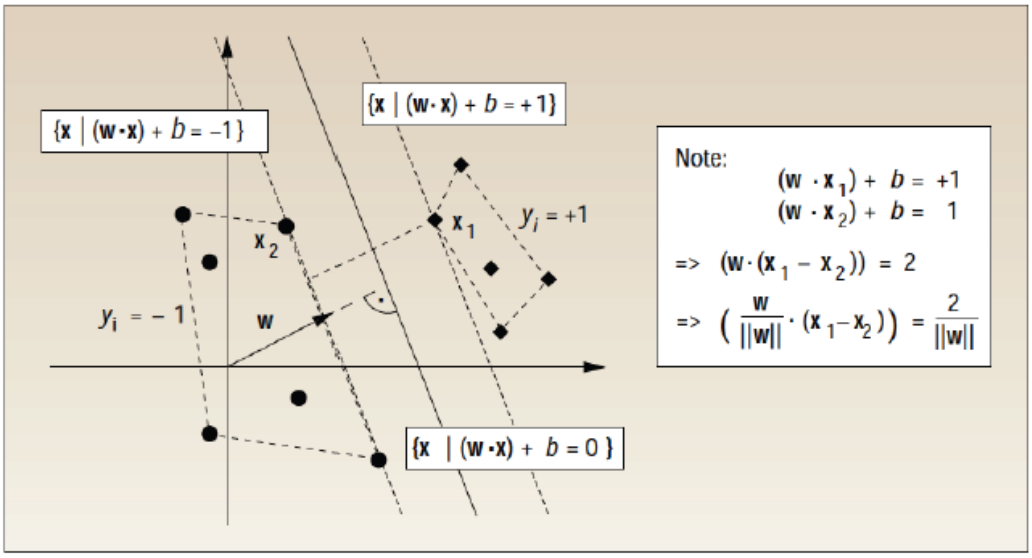


**Figure 3: SVM**
The instances on the dotted lines are the marginal instances (Dong et al., 2005).

For a linearly separable problem, the separating hyperplane can be expressed as a linear combination of marginal instances:

$$\boldsymbol{w} = \sum_i y_i a_i \boldsymbol{x_i} \qquad \textbf{Equation 7}$$

In the above formula, $\boldsymbol{w}$ is the separating hyperplane, $y_i \in \{+1, -1\}$ (binary problem), and $\boldsymbol{x_i}$ denotes the marginal instances which are called the 'support vectors'. On the other hand, according to figure 3, for positive examples $\boldsymbol{w}.\boldsymbol{x_i} \geq 1$ and for negative examples $\boldsymbol{w}.\boldsymbol{x_i} \leq -1$, so a new instance, $\boldsymbol{x}$, can be classified as:

$$class\ (\boldsymbol{x}) = sgn(\sum_i a_i \boldsymbol{x}.\boldsymbol{x_i} + b) \qquad \textbf{Equation 8}$$

For non-separable problems SVM adds additional dimensions to the original data, so that the new set becomes separable in the higher dimensional space. However, there is a trade-off between increasing the number of dimensions and a phenomenon called the *curse of dimensionality* i.e. the number of possible solutions grows exponentially by the number of variables. In addition, if SVM is applied on a version of data with too many dimensions it will overfit the data, in other words the hyperplanes will fit so well with the training data, but a new example could not be classified properly. On the other hand, as equation 8 shows, classifying a new instance requires computation of $\boldsymbol{x}.\boldsymbol{x_i}$ for every training example. In case of non-separable problem where $\boldsymbol{x_i}$ must be mapped to $\varphi(\boldsymbol{x_i})$

with more dimensions. So to label a new instance $\varphi(\boldsymbol{x}).\varphi(\boldsymbol{x_i})$ must be calculated which is computationally more expensive. The SVM solution to this problem is to use kernel functions $k(\boldsymbol{x}, \boldsymbol{x_i}) = \varphi(\boldsymbol{x}).\varphi(\boldsymbol{x_i})$ which has the same result but is cheap to compute. Therefore, in case of non-separable problems, the choice of the proper kernel function is key to success of SVM. Nevertheless most TC tasks have feature spaces with very high dimensions and are linearly separable (Joachims, 1998).

The original version of SVM handles binary problems, i.e. those containing two categories of data. However, there are a variety of approaches enabling it to generalize for multiclass problems as well. The simplest of these approaches is to train several classifiers in a one-versus-others fashion. More sophisticated approaches generalize the SVM optimization algorithm to handle multi-class problems (Noble, 2006).

Section Three

# Implementation

# 3. Implementation

The implementation of the project consisted of several phases which are mentioned in the following section.

## 3.1 Phases

1. Creating graph model for websites
    a. Graph object properties
    b. Parsing and adding web pages and links
2. Selecting topic related pages of the website
    a. Detecting the strongly connected component containing the home page
    b. Ranking and selecting best pages
3. Saving data from selected pages to the database
    a. Parsing Html source of web pages
    b. Saving unique tuples of (word, frequency, weight)
4. Creating a dataset of training and testing websites
    a. Choosing categories
    b. Selecting proper websites from each category
5. Selecting the features
    a. Computing IG and DFS of structure and content features
    b. Selection of features with top IG/DFS for use in classification
6. Weighting the features
    a. Calculating tf.idf of selected features for every website
    b. normalizing length of feature vectors
7. Training the classifier
    a. Creating topic vector files compatible with LIBSVM for train and test datasets
    b. Using ‘svm-train’ function to create a model of train data
    c. Using ‘svm-predict’ on test data to determine accuracy of classification
8. Classifying a new instance
    a. Creating topic vector for the new website
    b. Using ‘svm-predict’ to assign a label to it

## 3.2 Implementation in more detail

This project is implemented using a pc with dual core, 2.53 GHz CPU and 4.00 GB memory. The program is written with C# programming language in .NET (3.5) framework. The database is implemented using SQL Server 2008.

### 3.2.1 Graph Model

Every website is modeled as a directed graph in which each unique web page is considered as a node and every link between two nodes as a directed edge. A graph object has several properties among which there are 'domain', set of 'web pages' and 'topic related component'. In the current project it is supposed that the given URL by the user is the URL of home page of the website and the 'domain' property is derived from this URL. For instance if the given URL is:

'http://www.apistraining.com/apis/index.php'

Then the domain property will consist of two elements:

'www.apistraining.com'

'apistraining.com'

Any web page whose URL contains either of the above elements is considered as belonging to the website under consideration. The next property, 'web pages' or nodes of the graph are added by parsing the home page and following the links in a breadth-first fashion. The 'topic related component' consists of a list of web pages which are derived by analyzing the link structure and finding the strongly connected component that includes the home page.

The edges of the graph are implemented as a property of web page objects; each web page object has two lists for its inner and outer links which correspond to directed edges of the website graph. Given the URL of a website's homepage, the source Html is downloaded to find all the hyperlinks on the first page. The hyperlinks are added to a queue to be downloaded the same way as the first page in a breadth-first manner until there are no more unique web pages to download or the number of downloaded pages reaches a predefined limit.

### 3.2.2 Page Pruning

Among all the pages of a website only those which are most probably closely related to the topic of the website must be chosen to be further considered for classification. As mentioned in section two, in this study, first a strongly connected component of the graph is extracted, then a link based similarity measure is taken into account to choose a maximum of a predefined number of web pages as the topic related component of the graph.

To find the strongly connected component the Tarjan's cycle detection algorithm (Tarjan, 1972) is used. Then the pages of the main component, the strongly connected component with the biggest number of nodes, are ranked based on the ranking formula in Equation 2 and sorted according to their ranks. Then a maximum of 20 pages with the highest ranks are selected for being utilized in classification.

### 3.2.3 Processing and Saving Data from Websites

In this phase the textual content of the Html source of each page is downloaded with consideration of the Html tags embodying the text. In order to parse and get data from different nodes of the Html source of web pages, Html Agility Pack (Mourier & Klawiter, 2012), an Html parser implemented for .NET framework, is utilized. This code library models an Html document as a tree in which nodes are the Html tags, a DOM tree, and can handle real-world malformed Html as well.

The data that is saved consists of the text of 'title' of the web page, 'keywords' and 'description' attributes of meta tags, text content of paragraphs and headings, etc. that is displayed on the page, together with anchor texts which are displayed as the mouse is hovered over the hyperlinks. Meta data and simple text on the page compose the content structure of feature vector. During processing each word in content structure is saved together with a weight coefficient which is determined by the tag containing the word. This coefficient corresponds to the importance of the word; Meta data have the greatest importance and the next level belongs to words within heading tags and words with special format, i.e. bold, underlined or italic, respectively. In addition, anchor words, the structure features of the vector space, are given a unique and identical coefficient that signifies their type. For each website a word with its importance weight is considered as a unique entity; whenever the same entity is observed again its frequency is increased. In addition, each word in structure or content feature is reduced to its stem using Porter stemmer algorithm before being saved in the database.

### 3.2.4 Building the Dataset

For this prototype 5 website topics are chosen as different website categories to be examined. For each category 30 websites are downloaded to be utilized for training and 10 more for testing. Table 1 shows these categories:

**Table 1: Categories**

| 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|
| Education | Shopping | News | Health | Sport |

The websites are chosen from top categories of DMOZ ("DMOZ, open content directory,") directory service. Sample websites are chosen from different sub-categories of main categories. For this study only English websites are taken into account and for training those with multiple languages are discarded.

### 3.2.5 Dimensionality Reduction

Each website is modeled as a vector of a number of features that will be used by a learning algorithm. As noted in chapter 2.3.3, in text classification problems the number of features is very high, but many of them have little or no information value for classification and not only do not they benefit the classification but also act as noise, therefore they should be removed from the dataset. Two algorithms are utilized and compared in this prototype, Information Gain (IG) and Distinguishing Feature Selector (DFS). Structure and content features are considered as two separate sets while computing the value of IG and DFS. For every unique feature in each of these sets IG and DFS are calculated and saved according to equations 3, 4, respectively. Finally features are sorted by their value of IG/DFS and top $n\%$ of them is chosen for being used in classification, where $n$ is an arbitrary number corresponding to the factor by which the size of database is reduced.

### 3.2.6 Feature Weighting

Every website is modeled with a vector of features, each of which having a weight value that signifies the importance of that feature in the website under consideration. After the

features are selected each of them will be considered for weight computation. Equation 5 is first used to compute $tf.idf$ value for every feature of structure and content set separately. However, because some websites, especially those belonging to news or health groups, generally have bigger number of words and the word frequency in such websites are normally bigger, that will lead to them dominating the classification. Therefore, after $tf.idf$ is computed for every feature of a website, its value will be updated according to equation 6, in which every weight value, is divided by the length of the vector so that its length is normalized. In this way, the weight value of every feature will be representative of the *relative* significance of that feature in a specific website. Further, it is worth mentioning that normalization is carried out separately for structure and feature components of the feature vector.

### 3.2.7 Training the classifier

In this prototype SVM algorithm is used for the classification purpose. In this study LIBSVM (Chang & Lin, 2011) which is a library for efficient classification with SVM is used. LIBSVM is an implementation of SVM which is optimized as for training and classification time and can be efficiently applied for multi-class problems. In order to apply LIBSVM for classification data must be written in a text file with the specified format determined by the authors of the program. Currently this process is time consuming which must be optimized in the future. The 'svm-train' function contained in the library creates a model of the data and saves the model in a text file with '.model' suffix. After the model is built, it can be utilized by 'svm-predict', another function used for testing and predicting, together with test data or data belonging to a new website, to predict label for instances. Function 'svm-predict' renders accuracy as well as an output file containing predicted class labels. Both 'svm-train' and 'svm-predict' are available as executable files and can be used from windows command line interpreter.

In addition, it is worth mentioning that LIBSVM offers many kernels with flexible parameters. But as stated in (Hsu, Chang, & Lin, 2003) for data with very large feature spaces the best accuracy is achieved by the simple linear kernel. This fact is examined and verified in this study.

### 3.2.8 Classifying a new website

The current prototype is implemented as a simple website that by receiving a website 'URL' returns its predicted class value. The current prototype can handle cases which

belong to one of the predefined classes. When the URL is received all the steps mentioned above should be taken to create a graph of the website, select relevant pages, select and save features, assign weight to features, create LIBSVM compatible files for topic vector of the website, and finally use ‘svm-predict’ to predict the new website’s class label. To perform the last step, i.e. class label prediction, the open source code of LIBSVM is integrated into the prototype’s code. The result of classification is retrieved form an output file which is delivered by ‘svm-predict’.

Section Four

# Conclusion

# 4. Conclusion

## 4.1 Results

The dataset of training and test websites are manually chosen from Dmoz ("DMOZ, open content directory,") directory service. It consists of 5 classes of websites chosen from different subclasses under one topic. Currently the training set consists of 150 and the test set of 50 websites. Table 2 shows the distribution of websites in the dataset.

**Table 2: Dataset**

| Number of categories | Number of train websites | Number of test websites |
|---|---|---|
| 5 | 30*5 | 10*5 |

The complete feature set is achieved by considering all the unique content and structure features downloaded from the training websites. Table 3 shows the number of features in the training dataset before feature selection.

**Table 3: Features in the training set**

| Number of Content Features (without feature selection) | Number of Structure Features (without feature selection) |
|---|---|
| 39431 | 19993 |

In (Uysal & Gunal, 2012) they claim that DFS measure outperforms the IG metric for feature selection. However, for the present dataset IG has better results, as shown in table 4.

**Table 4: Comparison of feature selection methods**

| Feature Selection Method | Number of Selected Content Features | Number of Selected Structure Features | Accuracy |
|---|---|---|---|
| DFS | 4000 | 3000 | 92% |
| IG | 4000 | 3000 | 94% |

The results presented in table 5 confirm the statement in (Hsu et al., 2003) about linear kernel being optimal for datasets with large number of features. As shown in table 5 the effect of using radial basis kernel is a dramatic reduction of accuracy, from 94% to 20%.

**Table 5: Comparison of two kernel functions**

| Training Mode | Number of Features | Accuracy (IG) |
|---|---|---|
| Kernel: Radial basis function $e^{(-\gamma * \lvert u-v \rvert^2)}$, $\gamma = \frac{1}{n.features}$ | 7000 | 20% |
| Kernel: Linear $u' * v$ | 7000 | 94% |

According to the results displayed in table 6, the best performance is achieved when utilizing almost 12% of the structure and content features.

**Table 6: impact of number of selected features on accuracy**

| Number of structure features | Number of content features | Accuracy |
|---|---|---|
| 1000 | 2000 | 58% |
| 2000 | 4000 | 92% |
| **3000** | **4000** | **94%** |
| 4000 | 6000 | 94% |

## 4.2 Summary

The rapid growth of internet necessitates construction of structure on the web. Internet is used for many different purposes, all of which requiring some kind of data access. Two main ways of finding relevant data are applying search engines or utilizing directory services. Various studies have considered web page and website classification with the approach to improve search results or automate expansion of web directories.

However, knowledge about purpose of a website or web page can be useful in online advertising as well. One of the cheapest and most common types of online advertising is publication of banner ads on web pages. Currently, ad servers publish banner ads randomly, but making this process ‘intelligent’ can lead to more effective advertising campaigns. Having knowledge about purpose of websites and web pages can help in making ad publishing intelligent.

According to different studies on web page classification, text classification methods can be used to create taxonomy on web pages. However, a web page is different from a simple text because of having structural characteristics incorporated in its Html source code, which can be utilized in classification. In addition, it is shown that ‘neighboring’ and ‘co-cited’ pages are topically related to each other.

The findings from studies on web page classification can be used in website classification. In fact, the first attempt for categorizing websites has been an extension of web page classification, so that a whole website is regarded as a ‘super-page’ which is labeled with the same methods

used for web page classification; however, the accuracy of this method is very low. To improve the accuracy some studies considered requiring knowledge about web page labels which is not always practical or efficient. Finally, other approaches have been proposed that utilize data from the relationship between pages and their relative importance, as well as structural properties of websites to categorize them. Besides, classification of websites has been subject to several researches with two main goals, determining either 'functionality' or 'topic' of websites; the aim of this study is to address the second problem, i.e. creating a topical taxonomy over websites.

In the current study, a website is modeled as a directed graph. A two-phase page selection is executed on the graph to eliminate noisy data. On the first phase of page selection, the strongly connected component of the graph containing the home page is chosen. Further, the information about relationship between neighboring and co-cited pages as well as significance of pages in terms of the number of links they receive is used for measurement of similarity between them and calculation of page ranks. Finally, the pages with highest ranks are chosen to be applied in classification. After selection of topic-related pages, their structural and textual features are extracted and represented in a uniform 'topic vector' for each website. To reduce the high dimensionality of topic vector, two feature selection approaches are implemented and examined, 'Information Gain' (IG) and 'Distinguishing Feature Selector' (DFS). In order to assign weight to the selected features for each website, the 'tf-idf' weighting approach with consideration of layout properties of terms is used. Finally, Support Vector Machines (SVM) classification algorithm is employed for classification of feature vectors.

The dataset consists of 5 categories, 150 training- and 50 test-websites, with a uniform distribution over categories. The best accuracy, 94%, is achieved when the classifier is trained on 12% of the features that are selected using information gain algorithm.

## 4.3 Future Work

Pre-processing of text content of websites includes word stemming and stop-word removal which are both specific to language, therefore there needs to be a separate stop-word list and stemming algorithm for every language. Therefore, only English websites are considered in this study. However, a future improvement could be to enable the system to handle websites with multiple languages or a language other than English.

Furthermore, for the current program to be able to recognize websites which belong to none of the predefined categories, a separate training set of websites which belong to none of existing categories is required. As a future improvement this problem might be solved by another choice of classifier or combining e.g. a non-supervised method like clustering with the current supervised algorithm.

The same motivation of this project, also applies to web page classification; ad servers can benefit likewise from such taxonomy on web pages as well. Therefore, an

unsupervised primary clustering of web pages, which could also increase the accuracy of website classification, would enable ad servers to take advantage of tagged web pages to specialize their ad publication even further.

# 5. Bibliography


BING, SEARCH ENGINE. FROM http://www.bing.com/

BRIN, S., & PAGE, L. (1998). THE ANATOMY OF A LARGE-SCALE HYPERTEXTUAL WEB SEARCH ENGINE. *COMPUTER NETWORKS AND ISDN SYSTEMS, 30*(1), 107-117.

CHAKRABARTI, S., DOM, B., & INDYK, P. (1998). *ENHANCED HYPERTEXT CATEGORIZATION USING HYPERLINKS.* PAPER PRESENTED AT THE ACM SIGMOD RECORD.

CHAKRABARTI, S., JOSHI, M. M., PUNERA, K., & PENNOCK, D. M. (2002). *THE STRUCTURE OF BROAD TOPICS ON THE WEB.* PAPER PRESENTED AT THE PROCEEDINGS OF THE 11TH INTERNATIONAL CONFERENCE ON WORLD WIDE WEB.

CHANG, C.-C., & LIN, C.-J. (2011). LIBSVM: A LIBRARY FOR SUPPORT VECTOR MACHINES. *ACM TRANSACTIONS ON INTELLIGENT SYSTEMS AND TECHNOLOGY (TIST), 2*(3), 27.

DMOZ, OPEN CONTENT DIRECTORY. FROM http://www.dmoz.org/

DONG, B., QI, G., & GU, X. (2005). DOMAIN-SPECIFIC WEBSITE RECOGNITION USING HYBRID VECTOR SPACE MODEL *ADVANCES IN WEB-AGE INFORMATION MANAGEMENT* (PP. 840-845): SPRINGER.

DREZE, X., & HUSSHERR, F.-X. (2003). INTERNET ADVERTISING: IS ANYBODY WATCHING? *JOURNAL OF INTERACTIVE MARKETING, 17*(4), 8-23.

ESTER, M., KRIEGEL, H.-P., & SCHUBERT, M. (2002). *WEB SITE MINING: A NEW WAY TO SPOT COMPETITORS, CUSTOMERS AND SUPPLIERS IN THE WORLD WIDE WEB.* PAPER PRESENTED AT THE PROCEEDINGS OF THE EIGHTH ACM SIGKDD INTERNATIONAL CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING.

FORMAN, G. (2003). AN EXTENSIVE EMPIRICAL STUDY OF FEATURE SELECTION METRICS FOR TEXT CLASSIFICATION. *THE JOURNAL OF MACHINE LEARNING RESEARCH, 3*, 1289-1305.

GODSIL, C. D., ROYLE, G., & GODSIL, C. (2001). *ALGEBRAIC GRAPH THEORY* (VOL. 8): SPRINGER NEW YORK.

GOOGLE, SEARCH ENGINE. FROM http://www.google.com/

HSU, C.-W., CHANG, C.-C., & LIN, C.-J. (2003). A PRACTICAL GUIDE TO SUPPORT VECTOR CLASSIFICATION.

IAB, P. (2013). IAB INTERNET ADVERTISING REVENUE REPORT 2012 FULL-YEAR RESULTS: MARKET RESEARCH REPORT, INTERACTIVE ADVERTISING BUREAU (IAB) AND PRICEWATERHOUSECOOPERS (PWC). http://www.iab.net/media/file/IAB_Internet_Advertising_Revenue_Report_FY_2012_rev.pdf.

JOACHIMS, T. (1998). *TEXT CATEGORIZATION WITH SUPPORT VECTOR MACHINES: LEARNING WITH MANY RELEVANT FEATURES*: SPRINGER.

JONES, K. S. (1972). A STATISTICAL INTERPRETATION OF TERM SPECIFICITY AND ITS APPLICATION IN RETRIEVAL. *JOURNAL OF DOCUMENTATION, 28*(1), 11-21.

KRIEGEL, H.-P., & SCHUBERT, M. (2004). CLASSIFICATION OF WEBSITES AS SETS OF FEATURE VECTORS. *PROC. IASTED DBA*, 127-132.

KWON, O.-W., & LEE, J.-H. (2003). TEXT CATEGORIZATION BASED ON *K*-NEAREST NEIGHBOR APPROACH FOR WEB SITE CLASSIFICATION. *INFORMATION PROCESSING & MANAGEMENT, 39*(1), 25-44.

LINDEMANN, C., & LITTIG, L. (2007). *CLASSIFYING WEB SITES.* PAPER PRESENTED AT THE PROCEEDINGS OF THE 16TH INTERNATIONAL CONFERENCE ON WORLD WIDE WEB.

LINDEMANN, C., & LITTIG, L. (2011). CLASSIFICATION OF WEB SITES AT SUPER-GENRE LEVEL *GENRES ON THE WEB* (PP. 211-235): SPRINGER.

MENCZER, F. (2005). MAPPING THE SEMANTICS OF WEB TEXT AND LINKS. *INTERNET COMPUTING, IEEE, 9*(3), 27-36.
MOURIER, S., & KLAWITER, J. (2012). HTML AGILITY PACK (VERSION 1.4.6). RETRIEVED FROM http://htmlagilitypack.codeplex.com/
NOBLE, W. S. (2006). WHAT IS A SUPPORT VECTOR MACHINE? *NATURE BIOTECHNOLOGY, 24*(12), 1565-1567.
NOVAK, T. P., & HOFFMAN, D. L. (1997). NEW METRICS FOR NEW MEDIA: TOWARD THE DEVELOPMENT OF WEB MEASUREMENT STANDARDS. *WORLD WIDE WEB JOURNAL, 2*(1), 213-246.
PAGE, L., BRIN, S., MOTWANI, R., & WINOGRAD, T. (1999). THE PAGERANK CITATION RANKING: BRINGING ORDER TO THE WEB: STANFORD DIGITAL LIBRARY TECHNOLOGIES.
QI, X., & DAVISON, B. D. (2006). *KNOWING A WEB PAGE BY THE COMPANY IT KEEPS.* PAPER PRESENTED AT THE PROCEEDINGS OF THE 15TH ACM INTERNATIONAL CONFERENCE ON INFORMATION AND KNOWLEDGE MANAGEMENT.
QI, X., & DAVISON, B. D. (2009). WEB PAGE CLASSIFICATION: FEATURES AND ALGORITHMS. *ACM COMPUTING SURVEYS (CSUR), 41*(2), 12.
RIBONI, D. (2002). *FEATURE SELECTION FOR WEB PAGE CLASSIFICATION.* PAPER PRESENTED AT THE EURASIA-ICT 2002 PROCEEDINGS OF THE WORKSHOP.
SALTON, G., & BUCKLEY, C. (1988). TERM-WEIGHTING APPROACHES IN AUTOMATIC TEXT RETRIEVAL. *INFORMATION PROCESSING & MANAGEMENT, 24*(5), 513-523.
SEBASTIANI, F. (2002). MACHINE LEARNING IN AUTOMATED TEXT CATEGORIZATION. *ACM COMPUTING SURVEYS (CSUR), 34*(1), 1-47.
SUN, A., LIM, E.-P., & NG, W.-K. (2002). *WEB CLASSIFICATION USING SUPPORT VECTOR MACHINE.* PAPER PRESENTED AT THE PROCEEDINGS OF THE 4TH INTERNATIONAL WORKSHOP ON WEB INFORMATION AND DATA MANAGEMENT.
TARJAN, R. (1972). DEPTH-FIRST SEARCH AND LINEAR GRAPH ALGORITHMS. *SIAM JOURNAL ON COMPUTING, 1*(2), 146-160.
TIAN, Y.-H., HUANG, T.-J., & GAO, W. (2004). TWO-PHASE WEB SITE CLASSIFICATION BASED ON HIDDEN MARKOV TREE MODELS. *WEB INTELLIGENCE AND AGENT SYSTEMS, 2*(4), 249-264.
UYSAL, A. K., & GUNAL, S. (2012). A NOVEL PROBABILISTIC FEATURE SELECTION METHOD FOR TEXT CLASSIFICATION. *KNOWLEDGE-BASED SYSTEMS, 36*, 226-235.
YAHOO!, WEB DIRECTORY. FROM http://www.yahoo.com/
YANG, Y., SLATTERY, S., & GHANI, R. (2002). A STUDY OF APPROACHES TO HYPERTEXT CATEGORIZATION. *JOURNAL OF INTELLIGENT INFORMATION SYSTEMS, 18*(2-3), 219-241.
ZHANG, J.-B., XU, Z.-M., XIU, K.-L., & PAN, Q.-S. (2010). A WEB SITE CLASSIFICATION APPROACH BASED ON ITS TOPOLOGICAL STRUCTURE. *INTERNATIONAL JOURNAL ON ASIAN LANGUAGE PROCESSING, 20*(2), 75-86.